

This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

1. (Currently Amended) A method for operating a computer using object-based computer code, the method comprising:

invoking an event handler method configured to handle a static event, using a delegate, by calling another method of an instance of a class for which parameters passed to the other method are also passed to the event handler method, wherein the parameters passed to the other method and to the event handler method comprise a sender parameter identifying an event source and an event arguments parameter comprising a package of a plurality of event arguments, a parameter list of the other method having a same signature as a parameter list of the event handler method, wherein the other method references the event handler method, and the delegate contains a reference to the other method;

creating an invocation list associated with the other method, the invocation list specifying one or more event handler methods to be invoked; ~~and~~

dynamically altering contents of the invocation list; and

referencing the static event using a member access of the form  $E.M$ ,

wherein  $E$  denotes a type and  $M$  denotes the static event.

2. (Original) The method of claim 1, further comprising adding an event handler method to the invocation list during execution of the object-based computer code.

3. (Original) The method of claim 2, further comprising using an event accessor to add the event handler method to the invocation list during execution of the object-based computer code.

4. (Original) The method of claim 2, further comprising using an addition operator to add the event handler method to the invocation list during execution of the object-based computer code.

5. (Original) The method of claim 1, further comprising removing an event handler method from the invocation list during execution of the object-based computer code.

6. (Original) The method of claim 5, further comprising using an event accessor to remove the event handler method from the invocation list during execution of the object based computer code.

7. (Original) The method of claim 5, further comprising using a subtraction operator to remove the event handler method from the invocation list during execution of the object-based computer code.

8. (Canceled)

9. (Canceled)

10. (Canceled)

11. (Canceled)

12. (Canceled)

13. (Canceled)

14. (Canceled)

15. (Canceled)

16. (Canceled)

17. (Currently Amended) A method for operating a computer using object-based computer code, the method comprising:

invoking an event handler method configured to handle a static event, using a delegate, by calling another method, wherein the other method references the event handler method, and the delegate contains a reference to the other method;

passing a sender parameter identifying an event source and an event arguments parameter to the other method and to the event handler method, parameter lists of the event handler method and the other method having compatible signatures, wherein the events arguments parameter comprises a package of a plurality of event arguments;

creating an invocation list associated with the other method, the invocation list specifying one or more event handler methods to be invoked; ~~and~~

altering contents of the invocation list during execution of the object-based computer code; and

referencing the static event using a member access of the form  $E.M$ , wherein  $E$  denotes a type and  $M$  denotes the static event.

18. (Original) The method of claim 17, further comprising adding an event handler method to the invocation list during execution of the object-based computer code.

19. (Original) The method of claim 18, further comprising using an event accessor to add the event handler method to the invocation list during execution of the object-based computer code.

20. (Original) The method of claim 18, further comprising using an addition operator to add the event handler method to the invocation list during execution of the object-based computer code.

21. (Original) The method of claim 17, further comprising removing an event handler method from the invocation list during execution of the object-based computer code.

22. (Original) The method of claim 21, further comprising using an event accessor to remove the event handler method from the invocation list during execution of the object-based computer code.

23. (Original) The method of claim 21, further comprising using a subtraction operator to remove the event handler method from the invocation list during execution of the object-based computer code.

24. (Currently Amended) A computer-readable medium having stored thereon a plurality of computer-executable modules, the computer-executable modules comprising:

an event source module configured to issue a static an event; and  
an event handler module configured to  
invoke an event handler method, using a delegate, by calling another method of an instance of a class for which parameters passed to the other method are also passed to the event handler method, wherein the parameters passed to the other method and to the event handler method comprise a sender parameter identifying an event source and an event arguments parameter comprising a package of a plurality of event arguments, a parameter list of the other method having a same signature as a parameter list of the event handler method, wherein the other method references the event handler method, and the delegate contains a reference to the other method, reference the static event,  
create an invocation list associated with the other method, the invocation list specifying one or more event handler methods to be invoked, ~~and~~  
dynamically alter contents of the invocation list, and  
reference the static event using a member access of the form  $E.M$ , wherein  $E$  denotes a type and  $M$  denotes the static event.

25. (Original) The computer-readable medium of claim 24, wherein the event handler module is further configured to add an event handler method to the invocation list during execution of the object-based computer code.

26. (Original) The computer-readable medium of claim 25, wherein the event handler module is further configured to use an event accessor to add the event handler method to the invocation list during execution of the object-based computer code.

27. (Original) The computer-readable medium of claim 25, wherein the event handler module is further configured to use an addition operator to add the event handler method to the invocation list during execution of the object-based computer code.

28. (Original) The computer-readable medium of claim 24, wherein the event handler module is further configured to remove an event handler method from the invocation list during execution of the object-based computer code.

29. (Original) The computer-readable medium of claim 28, wherein the event handler module is further configured to use an event accessor to remove the event handler method from the invocation list during execution of the object-based computer code.

30. (Original) The computer-readable medium of claim 28, wherein the event handler module is further configured to use a subtraction operator to remove the event handler method from the invocation list during execution of the object-based computer code.

31. (Canceled)

32. (Canceled)

33. (Canceled)

34. (Canceled)

35. (Canceled)

36. (Canceled)

37. (Canceled)

38. (Canceled)

39. (Canceled)

40. (Currently Amended) A computer-readable medium having stored thereon a plurality of computer-executable modules, the computer-executable modules comprising:

an event source module configured to issue ~~an~~ a static event; and

an event handler module configured to

invoke an event handler method, using a delegate, by calling another method, wherein the other method references reference the event handler method, and the delegate contains a reference to the other method,

pass a sender parameter identifying an event source and an event arguments parameter to the other method and to the event handler method, parameter lists of the event handler method and the other method having compatible signatures, wherein the events argument parameter comprises a package of a plurality of event arguments,

create an invocation list associated with the other method, the invocation list specifying one or more event handler methods to be invoked, ~~and~~

alter contents of the invocation list during execution of the object-based computer code, and

reference the static event using a member access of the form  $EM$ , wherein  $E$  denotes a type and  $M$  denotes the static event.

41. (Original) The computer-readable medium of claim 40, wherein the event handler module is further configured to add an event handler method to the invocation list during execution of the object-based computer code.

42. (Original) The computer-readable medium of claim 41, wherein the event handler module is further configured to use an event accessor to add the event handler method to the invocation list during execution of the object-based computer code.

43. (Original) The computer-readable medium of claim 41, wherein the event handler module is further configured to use an addition operator to add the event handler method to the invocation list during execution of the object-based computer code.

44. (Original) The computer-readable medium of claim 40, wherein the event handler module is further configured to remove an event handler method from the invocation list during execution of the object-based computer code.

45. (Original) The computer-readable medium of claim 44, wherein the event handler module is further configured to use an event accessor to remove the event handler method from the invocation list during execution of the object-based computer code.

46. (Original) The computer-readable medium of claim 44, wherein the event handler module is further configured to use a subtraction operator to remove the event handler method from the invocation list during execution of the object-based computer code.

47. (Canceled)

48. (Canceled)

49. (Canceled)

50. (Canceled)

51. (Canceled)

52. (Canceled)

53. (Canceled)

54. (Canceled)

55. (Canceled)

56. (Currently Amended) An object-based programming system comprising a computer configured to:

invoke an event handler method configured to handle a static event, using a delegate, by calling another method, wherein the other method references the event handler method, and the delegate contains a reference to the other method;

pass a sender parameter identifying an event source and an event arguments parameter to the other method and to the event handler method, parameter lists of the event handler method and the other method having compatible signatures, wherein the events argument parameter comprises a package of a plurality of event arguments;

create an invocation list associated with the other method, the invocation list specifying one or more event handler methods to be invoked; ~~and~~

alter contents of the invocation list during execution of the object-based computer code; and

reference the static event using a member access of the form  $E.M$ , wherein  $E$  denotes a type and  $M$  denotes the static event.

57. (Original) The object-based programming system of claim 56, wherein the computer is further configured to add an event handler method to the invocation list during execution of the object-based computer code.

58. (Original) The object-based programming system of claim 57, wherein the computer is further configured to use an event accessor to add the event handler method to the invocation list during execution of the object-based computer code.

59. (Original) The object-based programming system of claim 57, wherein the computer is further configured to use an addition operator to add the event handler method to the invocation list during execution of the object-based computer code.



60. (Original) The object-based programming system of claim 56, wherein the computer is further configured to remove an event handler method from the invocation list during execution of the object-based computer code.

61. (Original) The object-based programming system of claim 60, wherein the computer is further configured to use an event accessor to remove the event handler method from the invocation list during execution of the object-based computer code.

62. (Original) The object-based programming system of claim 60, wherein the computer is further configured to use a subtraction operator to remove the event handler method from the invocation list during execution of the object-based computer code.

63. (Canceled)

64. (Canceled)

65. (Canceled)

66. (Canceled)

67. (Canceled)

68. (Canceled)

69. (Canceled)

70. (Canceled)

71. (Canceled)

72. (Currently Amended) An object-based programming system comprising a computer configured to:

invoke an event handler method configured to handle a static event, using a delegate, by calling another method of an instance of a class for which parameters passed to the other method are also passed to the event handler method, wherein the parameters passed to the other method and to the event handler method comprise a sender parameter identifying an event source and an event arguments parameter comprising a package of a plurality of event arguments, a parameter list of the other method having a same signature as a parameter list of the event handler method, wherein the other method references the event handler method, and the delegate contains a reference to the other method;

create an invocation list associated with the other method, the invocation list specifying one or more event handler methods to be invoked; ~~and~~

dynamically alter contents of the invocation list; and

reference the static event using a member access of the form  $E.M$ , wherein  $E$  denotes a type and  $M$  denotes the static event.

73. (Original) The object-based programming system of claim 72, wherein the computer is further configured to add an event handler method to the invocation list during execution of the object-based computer code.

74. (Original) The object-based programming system of claim 73, wherein the computer is further configured to use an event accessor to add the event handler method to the invocation list during execution of the object-based computer code.

75. (Original) The object-based programming system of claim 72, wherein the computer is further configured to use an addition operator to add the event handler method to the invocation list during execution of the object-based computer code.

76. (Original) The object-based programming system of claim 72, wherein the computer is further configured to remove an event handler method from the invocation list during execution of the object-based computer code.

77. (Original) The object-based programming system of claim 76, wherein the computer is further configured to use an event accessor to remove the event handler method from the invocation list during execution of the object-based computer code.

78. (Original) The object-based programming system of claim 76, wherein the computer is further configured to use a subtraction operator to remove the event handler method from the invocation list during execution of the object-based computer code.

79. (Canceled)

80. (Canceled)

81. (Canceled)

82. (Canceled)

83. (Canceled)

84. (Canceled)

85. (Canceled)

86. (Canceled)

87. (Canceled)

**REMARKS/ARGUMENTS**

The Examiner contacted the undersigned attorney via telephone on February 16, 2005. The Examiner suggested that the independent claims would be allowable if amended to include similar features to those recited in dependent claims 8 and 9. Applicants have consequently amended the independent claims 1, 17, 24, 40, and 56 to include these features. Applicants have canceled claims 8-15, 31-38, 47-54, 63-71, and 79-86. No new matter has been added.

**CONCLUSION**

Applicant respectfully requests reconsideration of the claims and early issuance of a Notice of Allowance.

Date: February 17, 2005

/Jonathan M. Waldman/  
Jonathan M. Waldman  
Registration No. 40,861

Woodcock Washburn LLP  
One Liberty Place - 46th Floor  
Philadelphia PA 19103  
Telephone: (215) 568-3100  
Facsimile: (215) 568-3439